**Homework: manual test generation for grep**

1. (100 points) Generate a set of test cases for the attached grep v2.0 that achieves high statement and branch coverage.
   1. Read online grep manual carefully to understand the functionality of grep.
      * Note that some options for the latest grep version may not work for grep v2.0
   2. Download grep\_v2.0.zip from the class web page and compile it by gcc –coverage grep.c –o grep20
   3. Create (**at least 10**) test cases to exercise different options, patterns, and target files to cover as many statements and branches as you can.
   4. Measure both statement coverage (i.e., lines executed) and branch coverage of the created test cases by using gcov. Your test cases must achieve more than 50% branch coverage.
   5. Identify the functions which are not covered at all (to improve coverage further)

You should submit the following items so that TA can replay your testing result completely (any missing information will reduce score).

1. The text file test\_script which contains grep commands (or script to generate grep commands) each of which corresponds to your test case
   * + Ex.

grep20 -B 5 -n [a-z][0-9][a-z] abc.c

grep20 000 abc.txt

…

* + - For realistic testing experience in practice (under time pressure and computing resource pressure), test\_script should complete in less than 1 minute and a target file such as abc.txt should be less than 100 Kbytes long.

1. A target text files which are used by test\_script (submit only softcopy)

* Ex. abc.c, abc.txt

1. A total number of the statements and the statement coverage achieved by the test cases
2. A total number of the branches and the branch coverage achieved by the test cases

* You have to report the number of **taken branches**

1. The **“best” (i.e. the most effective) test case** which achieves the largest branch coverage among the test cases you created and its branch coverage achieved.
2. A list of the **3 test cases** that achieve the largest branch coverage altogether and the accumulated branch coverage achieved by the 3 test cases (known as *test case prioritization*).
3. A list of functions which are not executed at all by your test cases.
4. The number of branches for every unexecuted function
5. A total number of the branches and the branch coverage achieved by the test cases for regex\_compile()
6. (Extra 50 points) If you achieve the 1st/or 2nd highest branch coverage by the selected 3 test cases among your classmates
7. (Extra 100 points) Find and explain any NEW bug in grep (i.e. a bug that have not been known before publicly. For example, you have to search [**http://lists.gnu.org/archive/html/bug-grep/**](http://lists.gnu.org/archive/html/bug-grep/)). To avoid redundant bug reports, report the bug you have detected at KLMS asap. Only the first one who reports a bug at KLMS will get 100 points per bug.